Machine Learning for biology

V. Monbet

UFR de Mathématiques
Université de Rennes 1

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

This part of the course is highly inspired from J.P. Vert lecture notes/slides.

Outline

Kernel methods

Motivations

- Develop versatile algorithms (based on pairwise comparison) to process and analyze data
- without making any assumptions regarding the type of data (vectors, strings, graphs, images, ...)

The approach

- Develop methods based on pairwise comparisons

Kernel methods

Representation of pairwise comparison, ideas

- Define a "comparison function": $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$
- Represent a set of n data points $\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ by the $n \times n$ matrix (symmetric and positive semidefinite): $[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$.
- Example

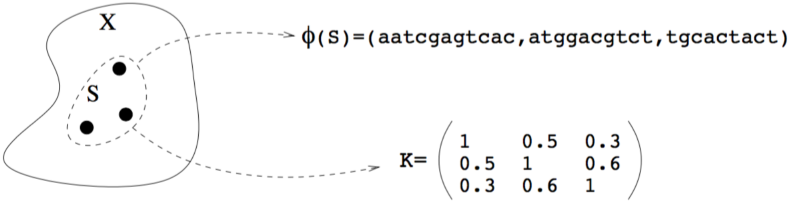
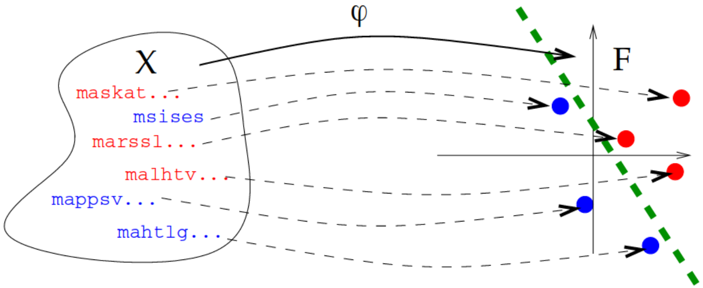


Figure from J.P. Vert

Supervised classification with vector embedding

- Map each string $x \in \mathcal{X}$ to a vector $\varphi(x) \in F$.
- Train a classifier for vectors on the images $\varphi(x_1), \cdots, \varphi(x_n)$ of the training set (nearest neighbor, linear perceptron, logistic regression,support vector machine...)



Kernel trick

- In statistics, most methods are not directly based on the variables x or $\varphi(x)$ itself but on their inner product
$$(x,y) \mapsto K(x,y) = \langle \varphi(x), \varphi(y) \rangle$$
because the inner product describes the general geometrical structure of the set.
- The kernel trick consists in forgetting the transformation φ and directly use the kernel.

Definition

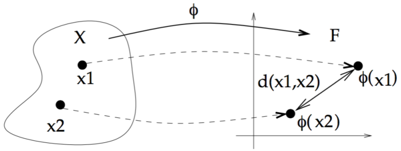
A positive definite (p.d.) kernel on $\mathcal{X}$ is a function
$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
such that for every sequence x_i of points in $\mathcal{X}$ the matrix $(k(x_i, x_j))_{i,j}$ is symetric and positive.

Kernel trick

- The trick: this mapping might not be explicitly given.
- Example: computing distances

$$
\begin{aligned}
d_K(\mathbf{x}_1,\mathbf{x}_2)^2 &= ||\varphi(\mathbf{x}_1) - \varphi(\mathbf{x}_2)||^2_{\mathcal{X}} \\
&= \langle \varphi(\mathbf{x}_1) - \varphi(\mathbf{x}_2), \varphi(\mathbf{x}_1) - \varphi(\mathbf{x}_2)\rangle_{\mathcal{X}} \\
&= \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_1)\rangle_{\mathcal{F}} + \langle \varphi(\mathbf{x}_2), \varphi(\mathbf{x}_2)\rangle_{\mathcal{F}} - 2\langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2)\rangle_{\mathcal{F}} \\
d_K(\mathbf{x}_1,\mathbf{x}_2)^2 &= K(\mathbf{x}_1,\mathbf{x}_1) + K(\mathbf{x}_2,\mathbf{x}_2) - 2K(\mathbf{x}_1,\mathbf{x}_2)
\end{aligned}
$$

where K is a kernel.



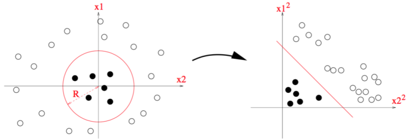
Example of Kernels

- Let $\mathcal{X} = \mathbb{R}^p$.
  The function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ defined by $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle_{\mathbb{R}^p}$ is a p.d. kernel.
  It is linear.
- Let $\mathcal{X}$ be any set, and $\varphi : \mathcal{X} \mapsto \mathbb{R}^d$.
  Then, the function : $\mathcal{X}^2 \mapsto \mathbb{R}$ defined as follows is p.d kernel.

$$\forall\, (\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, \quad K(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathbb{R}^p}.$$

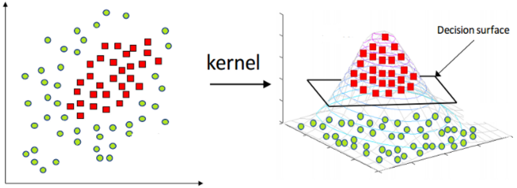
- Example: polynomial kernel.
  For $\mathbf{x}^T = (x_1, x_2)$, $\varphi(\mathbf{x}) = \{x_1^2, \sqrt{2} x_1 x_2, x_2^2\}$,

$$K(\mathbf{x}, \mathbf{x}') = x_1^2 x_1'^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 x_2'^2 = \langle \mathbf{x}, \mathbf{x}' \rangle_{\mathbb{R}^2}^2$$



The transformation leads to a linear separation problem!

Polynomial kernel



- Here the idea is to map the data into a (possibly high dimensional) vector space where linear relations exist among the data, then apply a linear algorithm in this space.
- Problem: Representing data in a highdimensional space is computationally difficult
- Alternative solution to the original problem: Calculate a similarity measure in the feature space instead of the coordinates of the vectors there, then apply algorithms that only need the value of this measure.

Outline

Mercer's condition

- For which kernels does there exist a pair $(\mathcal{H}, \varphi)$ where $\mathcal{H}$ is a (possibly infinite dimensional) Euclidean space and $\varphi : \mathbb{R}^p \mapsto \mathcal{H}$ is the mapping?
- Mercer's condition tells us whether or not a prospective kernel is actually a dot product in some space.
- It can be stated as follows:
  There exists a mapping φ and an expansion

  $$K(x, y) = \sum_{i=1}^{n} \varphi_i(x)\varphi_i(y)$$

  if and only if for any $g(x)$ such that $\int g(x)^2 dx$ is finite then $\int K(x, y)g(x)g(y)dxdy \geq 0$ (in otherwords, K is semi-positive definite.
- It can be shown that this condition is satisfied for positive integral powers the dot product:

  $$K(x, y) = (x \cdot y)^d$$

Constructing a feature space

- Given that we want a kernel function K that satisfies $K(x, y) = \langle \varphi(x), \varphi(y) \rangle$, how do we construct a feature space for K?
- 1. Define a feature map
$$\varphi : \mathcal{X} \to \mathbb{R}^n, \quad x \mapsto K(., x)$$
Then $\varphi(x) = K(., x)$ denotes the function that assigns the value $K(x', x)$ to $x' \in \mathcal{X}$.
- 2. Turn it into a linear space
$$f(.) = \sum_{i=1}^{m} \alpha_i K(., x_i), \quad g(.) = \sum_{i=1}^{m'} \beta_j K(., x_j')$$
- 3. Endow it with a dot product
$$\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x_j')$$
and turn it into an Hilbert space[1] $\mathcal{H}$.

[1] An Hilbert space is a vector space where an inner product is defined.

Outline

Linear kernel

- Take $\mathcal{X} = \mathbb{R}^d$ and the linear kernel:

$$K(x, y) = \langle x, y \rangle_{\mathbb{R}^d}.$$

Theorem

The RKHS of the linear kernel is the set of linear functions of the form

$$f_w(x) = \langle w, x \rangle_{\mathbb{R}^d} \text{ for } w \in \mathbb{R}^d$$

endowed with the inner product

$$\forall w, v \in \mathbb{R}^d, \langle f_w, f_v \rangle_{\mathcal{H}} = \langle w, v \rangle_{\mathbb{R}^d}$$

and corresponding norm

$$\forall w \in \mathbb{R}^d, ||f_w||_{\mathcal{H}} = ||w||_{\mathbb{R}^d}$$

Non linear kernels

- Take $\mathcal{X} = \mathbb{R}^d$ and the polynomial kernel:

$$K(x, y) = (a\langle x, y\rangle_{\mathbb{R}^d} + 1)^{\delta}$$

with δ the degree of the polynom.

- Take $\mathcal{X} = \mathbb{R}^d$ and Gaussian kernel

$$K(x, y) = e^{-\frac{||x-y||^2}{\sigma^2}}.$$

with σ the width of the kernel. .

Example of string Kernels

- Les protéines sont des chaines d'acides aminés qui diffèrent selon leur longueur et leur composition.
- Exemples de longueur 110 et 153. L'alphabet contient 20 caractères.

  IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEEIFQGIGTLESQTVQGGTV
  ERLFKNLSLIKKYIDGQKKKCGEERRRVNQFLDYLQEFLGVMNTEWI

  PHRRDLCSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAERLQENLQAYRTFHVLLA
  RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGML
  FEKKLWGLKVLQELSQWTVRSIHDLRFISSHQTGIP

- Il existe plusieurs façons de mesurer la similarité entre les deux molécules.
- Spectral-kernel. On considère une mesure basée de le nombre d'occurrence de sous séquences (ex : LQE).
  Pour construire les variables, on compte le nombre dâĂŹoccurrences de toutes les séquences de longueur m. On génère ainsi de nouvelles variables pour lesquelles on peut définir des noyaux.
- On pourrait aussi compter le nombre de positions communes ou le nombre de sous séquences communes (en autorisant éventuellement des gaps).

Outline

Kernel PCA

- Consider the PCA of a set of transformed individuals $\varphi(\mathbf{x}_i)$, $i \in \{1, \dots, n\}$.
- Lets φ be the matrix with ist line $\varphi(\mathbf{x}_i)$.
- The Singular Value Decomposition[2] of

$$\varphi\varphi^T = \left(k(\mathbf{x}_i, \mathbf{x}_j)\right)_{ij}$$

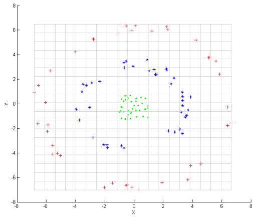
  returns the principal components (eigen vectors $\alpha_1, \cdots$ of the matrix with coefficients $k(\mathbf{x}_i, \mathbf{x}_j)$).
- The coordinates of a new sample $\mathbf{x}'$ are given by the inner product $\langle k(\mathbf{x}', \mathbf{x}_i)_i, \alpha_j \rangle$.

+ Avantage: the reconstruction space is not explicitly needed, but only the kernel $k(.,.)$.
- Drawback: the computation of the coordinates depends on the number of observations in the learning set.

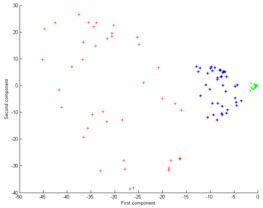
[2] SVD = looking for the eigen values and eigen vectors as in PCA. It leads to principal conponents.

Exemple jouet

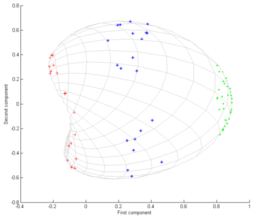
Initial set



Polynomial kernel



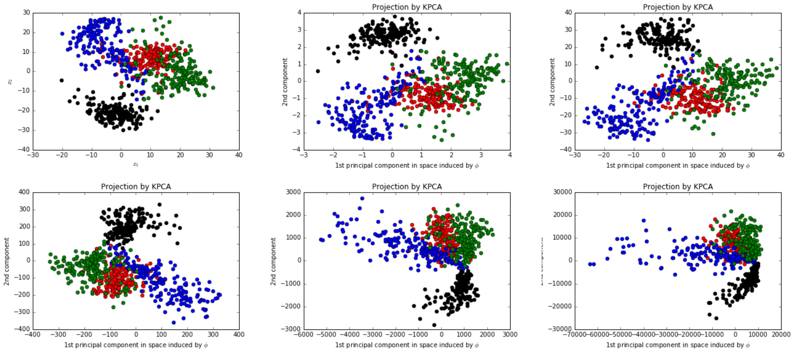
Gaussian kernel



$$k(x,x') = (x^T x' + 1)^2, \quad k(x,x') \propto e^{-\frac{1}{2}(x-x')^T(x-x')}$$

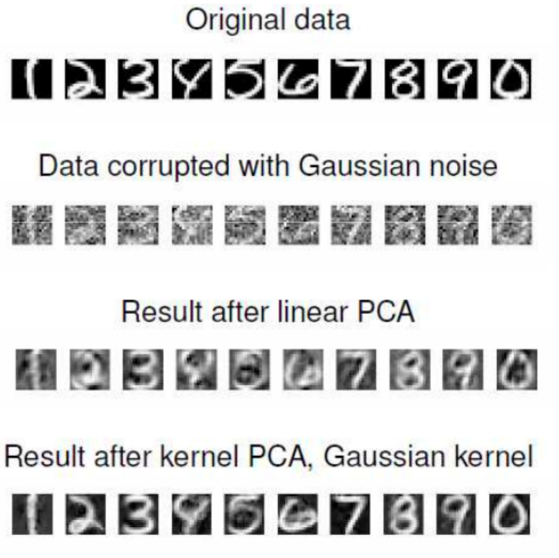
Digits

Projection of digits 0 to 3 (black, blue, red, green) for the regular PCA (top left panel) and kernel ACP with polynomial kernel of degree 1 to 5.

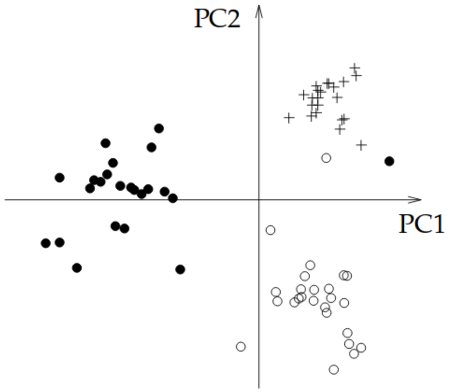


The space is transformed. However it seems difficult here to decide if one transformation is better thatn the others.

Digits


Original data
Data corrupted with Gaussian noise
Result after linear PCA
Result after kernel PCA, Gaussian kernel

tRNA sequences



A set of 74 human tRNA sequences is analyzed using a kernel for sequences (the second-order marginalized kernel based on SCFG). This set of tRNAs contains three classes, called Ala-AGC (*white circles*), Asn-GTT (*black circles*) and Cys-GCA (*plus symbols*) (from Tsuda et al., 2003).

Outline

Kernel k-means

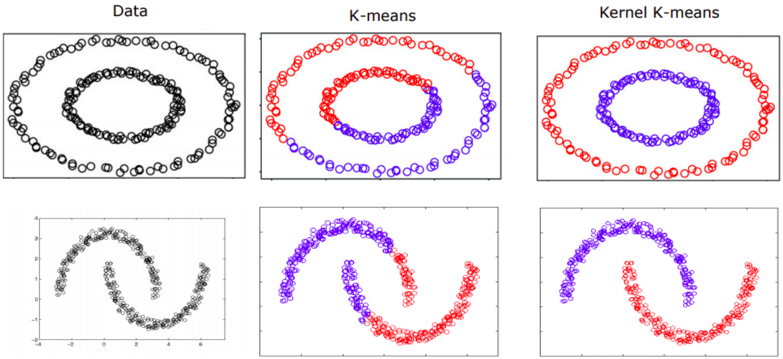
Kernel kmeans can be used to detect non convex clusters

- kmeans is known to only detect cluster that are linearly separable.
- Idea: project the data into a space F where the clusters are linearly separable.
- Drawback: the computation will be more expensive.
- Kernel kmeans minimizes the SSE

$$\sum_{k=1}^{K} \sum_{i \in C_k} ||\varphi(x_i) - \mu_k^{(F)}||_{\mathcal{H}}^2 \text{ where } \mu_k^{(F)} = \frac{1}{\text{card}(C_k)} \sum_{i \in C_k} \varphi(x_i)$$

It can be shown after short calculations that

$$||\varphi(x_i) - \mu_k^{(F)}||_{\mathcal{H}}^2$$
$$= \left(K(x_i, x_i) - \frac{2}{\text{card}(C_k)} \sum_{j \in C_k} K(x_i, x_j) + \frac{1}{\text{card}(C_k)^2} \sum_{j \in C_k} \sum_{\ell \in C_k} K(x_j, x_\ell) \right)$$



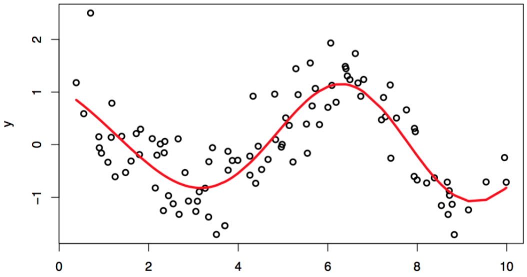
Kernel K-means is able to find "complex" clusters.

Outline

Kernel Ridge Regression

- $\mathcal{S}_n = \{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,n}$ a training set
- Goal = find a function f to predict y by $f(\mathbf{x})$
- Least-square regression with penalization to prevent overfitting

$$\hat{f} = arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda ||f||^2_{\mathcal{F}}$$



Kernel Ridge Regression

$$\hat{f} = arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda ||f||_{\mathcal{F}}^2$$

- By the representer theorem, any solution can be expanded as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

- Let $\mathbf{K}$ be a $n \times n$ Gram matrix: $\mathbf{K}_{ij} = K(x_i, x_j)$
- We can then write: $(\hat{f}(\mathbf{x}_1), \cdots, \hat{f}(\mathbf{x}_n))^T = \mathbf{K}\boldsymbol{\alpha}$
- The following holds:

$$||\hat{f}||_{\mathcal{F}}^2 = \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x_k}) \alpha_k = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$$

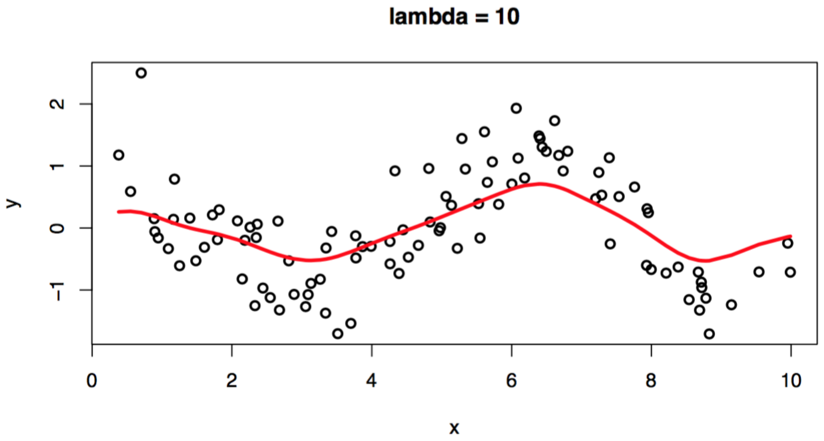
- The kernel Ridge regression problem is therefore equivalent to

$$\arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} (\mathbf{K}\boldsymbol{\alpha} - y)^T (\mathbf{K}\boldsymbol{\alpha} - y) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$$

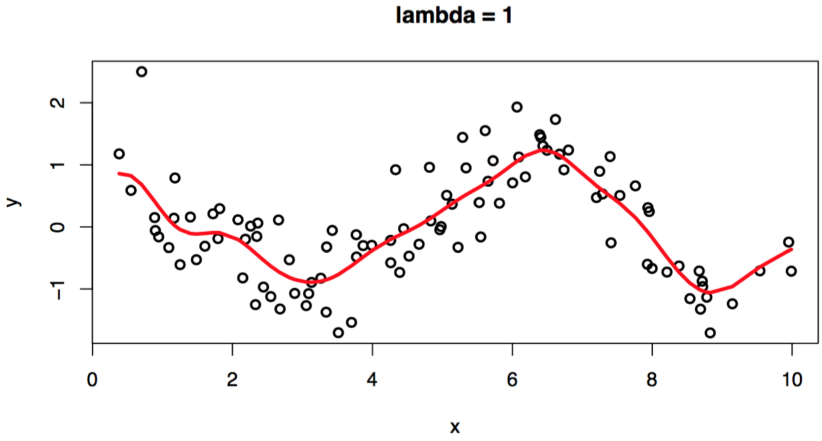
and its solution is

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda n \mathbf{I})^{-1} y$$

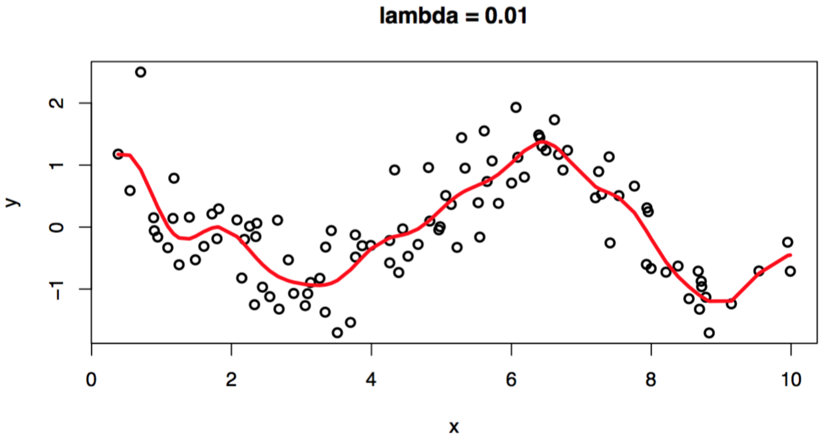
Example with Gaussian kernel



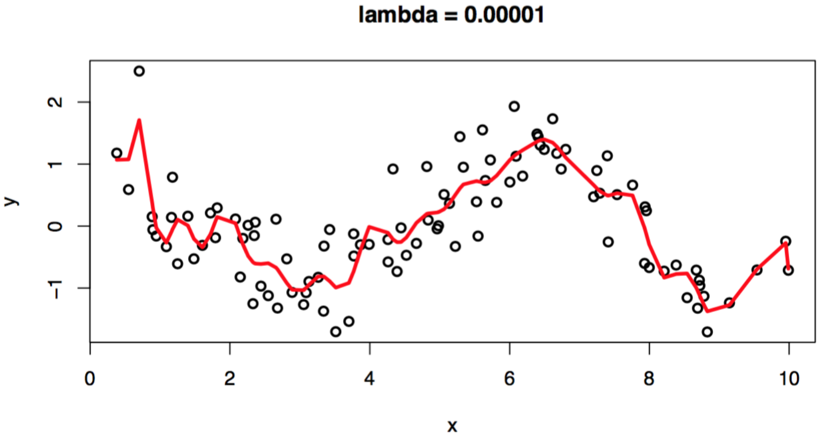
Example with Gaussian kernel



Example with Gaussian kernel



Example with Gaussian kernel



- The kernel trick allows to extend many linear algorithms to non-linear settings and to general data (even non-vectorial).
- The representer theorem shows that functional optimization over (subsets of) the kernel space is feasible in practice.
- We will see next a particularly successful applications of kernel methods: supervized classification with SVM.